# COMPUTING THE BRST OPERATOR USED IN QUANTIZATION OF GAUGE THEORIES

A. BURNEL, H. CAPRASSE*
*Département d'Astronomie et d'Astrophysique, Université de Liège*
*Institut de Physique (B5), B-4000 Liège I, Belgium*

and

A. DRESSE+
*Université libre de Bruxelles, CP 210/01 Blvd du Triomphe*
*B-1050 Bruxelles*

ABSTRACT

It is shown that for a large class of non-holonomic quantum mechanical systems one can make the computation of BRST charge fully algorithmic. Two computer algebra programs written in the language of `REDUCE` are described. They are able to realize the complex calculations needed to determine the charge for general nonlinear algebras. Some interesting specific solutions are discussed.

*Keywords*: Gauge Theory; Computer Algebra.

## 1. Introduction

BRST theory[1] has proven to be a very powerful tool to treat gauge theories and, in particular, to quantize gauge field theories. Indeed it may be useful to get the physical states from a space larger than seems necessary, because the restriction of the phase space to the physically meaningful configurations often hides the natural symmetries. Some systems, such as Yang-Mills theory, do not even admit global gauge fixing conditions. In such cases BRST theory naturally introduces the ghosts, necessary to write the path integral formulation of gauge theories.

At the core of BRST theory is the BRST charge $Q_B$, used to select the physical states from a larger space, and specify the physical equivalence of two a priori different states.

$$Q_B|\Psi_{phys}>=0, \tag{1.1}$$

$$|\Psi_{phys}> \quad \approx \quad |\Psi'_{phys}> + Q_B|\Psi_0> . \tag{1.2}$$

*E-mail: caprasse@vm1.ulg.ac.be
+E-mail: adresse@ulb.ac.be

The BRST charge is closely related to the symmetries of the system described in the Lagrangian formulation through Noether theorem. Its construction can, however, be based only on the knowledge of the algebra of constraints.

A recursive procedure to build the BRST operator order by order in the ghosts has been developped but that procedure is not fully algorithmic. We show here that the procedure can be made algorithmic for an important class of theories. We describe the capabilities of two programs written in `REDUCE`[2] which do indeed allow to compute the corresponding BRST charge. This class covers quadratic algebras such as those discussed in ref. 3,4, thus extending the realm of application of BRST theory beyond usual gauge theories. The use of computer algebra is mandatory for all but the most simple cases as the computations involved are tedious, although systematic.

The paper is organized as follows.
In sec. 2, the construction of the BRST operator is briefly described and two algorithms for its construction are given. The programs are described in sec. 3 while their applications are illustrated in sec. 4. Conclusions as well as the present limitations of these programs are presented in sec. 5.

## 2. The BRST operator

In the Hamiltonian formulation of a gauge theory, the symmetries of the system are described by the constraints, which are relations between the generalized coordinates and the momenta. The constraints of the Hamiltonian formulation are not all related to gauge invariance. Those related to it are called first class constraints. The others, said to be of second class, can be eliminated by an adequate redefinition of the brackets.

Let us consider a set of $N$ first class constraints

$$\{C_a, \quad a = 1, \ldots, N\}.$$

They satisfy a Poisson bracket algebra

$$\{C_a, C_b\} = f_{ab}{}^c C_c \tag{2.1}$$

where the f's, the structure functions depend on the canonical variables. Here and in the following, summation over repeated indices is understood except otherwise stated.

The computation of the BRST charge can be made fully algorithmic if we assume that one can write the structure functions in terms of the constraints. Eq (2.1) can then be written as

$$\{C_a, C_b\} = \sum_{i=1}^{q} f_{i\,ab}^{a_1 \ldots a_i} \prod_{j=1}^{i} C_{a_j} \tag{2.2}$$

where, now, the $f_i$'s are constants and $q$ is a number obtained from the relations between the constraints and the structure functions. In the present work, the necessary manipulations to obtain it are not considered so that Eq. (2.2) is the starting point.

To each constraint $C_a$, one associates two variables of odd Grassmann parity : a ghost $\eta^a$ with ghost number 1 and a ghost momentum $\mathcal{P}_a$ with ghost number -1. They satisfy the following bracket algebra[a].

$$\{\eta^a, \eta^b\} = \{\mathcal{P}_a, \mathcal{P}_b\} = 0, \quad \{\eta^a, \mathcal{P}_b\} = -\delta^a_b. \tag{2.3}$$

The BRST operator $\Omega$ is defined by

- its nilpotency ($\{\Omega, \Omega\} = 0$),
- its ghost number 1
- it involves the term $\eta^a C_a$

It can be decomposed in powers of the ghosts as

$$\Omega = \sum_{n=0}^{N-1} \Omega^{(n)} = \sum_{n=0}^{N-1} A^{j_1 \dots j_n}_{i_1 \dots i_{n+1}} \eta^{i_1} \dots \eta^{i_{n+1}} \mathcal{P}_{j_1} \dots \mathcal{P}_{j_n} \tag{2.4}$$

where the $A$'s depend only on the constraints and structure constants of the problem. It can be determined in a recursive way by

$$\Omega^{(0)} = \eta^a C_a, \tag{2.5}$$

$$\delta\Omega^{(n+1)} = -D^{(n)} \tag{2.6}$$

with

$$\delta F = -F \frac{\overleftarrow{\partial}}{\partial \mathcal{P}_a} C_a \tag{2.7}$$

and

$$D^{(n)} = \frac{1}{2} \left( \sum_{k=0}^{n} \{\Omega^{(k)}, \Omega^{(n-k)}\}_C + \sum_{k=0}^{n-1} \{\Omega^{(k+1)}, \Omega^{(n-k)}\}_{gh} \right). \tag{2.8}$$

The second term is not present when $n = 0$. Here, $\{\cdot, \cdot\}_C$ and $\{\cdot, \cdot\}_{gh}$ are respectively the constraint and ghost Poisson brackets.

The computable $D$ is written in the form

$$D^{(n-1)} = Z^{(a)a_1 \cdots a_{n-2} b} C_{(a)} \mathcal{P}_{a_1} \cdots \mathcal{P}_{a_{n-2}} C_b \tag{2.9}$$

in order to make the computation algorithmic. The $Z$'s depend on the ghosts $\eta^a$ and structure constants and are antisymmetric over the $a_1, \cdots, a_{n-2}, b$. $(a)$ is a set of indices which may be empty. Note that this expression is not easy to obtain explicitly, as it involves a decomposition of a polynomial ($D^{(n-1)}$) in products of the constraints, seen as generators of a polynomial ideal.

From Eq. (2.4), one writes

$$\Omega^{(n)} = K^{(a)a_1 \cdots a_{n-1}} C_{(a)} \mathcal{P}_{a_1} \cdots \mathcal{P}_{a_{n-1}} \tag{2.10}$$

[a] The choice of sign in the bracket of ghosts and ghost momenta is conventional. In this paper, the conventions of ref. 1 are adopted.

where $K^{(a)a_1\cdots a_{n-1}}$ is antisymmetric over $a_1,\cdots,a_{n-1}$. $C_{(a)}$ denotes the product of constraints whose indices are in the set $(a)$.

Eq. (2.6) leads to

$$(n-1)K^{(a)a_1\cdots a_{n-2}b}C_{(a)}\mathcal{P}_{a_1}\cdots\mathcal{P}_{a_{n-2}}C_b = Z^{(a)a_1\cdots a_{n-2}b}C_{(a)}\mathcal{P}_{a_1}\cdots\mathcal{P}_{a_{n-2}}C_b. \tag{2.11}$$

Its solution is

$$K^{(a)a_1\cdots a_{n-2}b} = \frac{1}{n-1}\left(Z^{(a)a_1\cdots a_{n-2}b} + G^{(a)a_1\cdots a_{n-2}b}\right) \tag{2.12}$$

as Z is antisymmetric as mentionned.

The tensor $G$ is arbitrary except that it is antisymmetric over the set of indices formed by the union of $a_1,\cdots,a_{n-2},b$ and one index of the set $(a)$. In practice, it will be built from the tensor $Z$ with the above antisymmetrisation, each possible term being multiplied by an arbitrary coefficient.

Since the BRST charge is defined only up to an arbitrary BRST exact term, there exist infinitely many solutions of Eq. (2.6). In particular, the $\Omega^{(n)}$ can be built as an arbitrary linear combination of the type

$$\sum_{i=0}^{n-1}\alpha_i Z^{a_1\ldots a_{i-1}(a)a_i\ldots a_{n-1}} \tag{2.13}$$

where no antisymmetrisation on the indices of $Z$ is involved. Some of the coefficients $\alpha_i$ will be fixed by imposing Eq.(2.6). It is left to find heuristic criteria to fix the remaining ones. Two of them are explained in the next section together with the corresponding programs.

## 3. Description of the Programs

A preliminary step to the writing of the programs is to find ways to fix all constants $\alpha_i$'s. This is done in two ways. They are successively described.

In the first one, during the calculation of each $\Omega^{(n)}$, the coefficients $\alpha_i$'s are chosen to minimize the number of terms.

In the second one, one assumes that the BRST charge remains invariant under a redefinition of the constraints, of the ghosts and the ghost momenta. This kind of constraint is usual in homological perturbation theory. The principles at the basis of this choice are the following:

Suppose one can define a linear operator $\sigma$, and a partition $\mathcal{A} = \oplus_k \mathcal{A}_k$ of the algebra of the constraints such that, for all $n$,

$$(\delta\sigma + \sigma\delta)A = N_n A \tag{3.1}$$

for all $A \in \mathcal{A}_n$. Such an operator is called a contracting homotopy operator for $N$.

Now consider the equation (2.6) where $D^{(n)}$ is known, and $\Omega^{(n+1)}$ has to be determined. Decompose them as

$$\Omega^{(n+1)} = \sum_i \Omega_i^{(n+1)} \qquad \Omega_i^{(n+1)} \in \mathcal{A}_i, \tag{3.3}$$

$$D^{(n)} = \sum_i D_i^{(n)} \qquad D_i^{(n)} \in \mathcal{A}_i. \tag{3.4}$$

$\delta D_i^{(n)} = 0$ since $\delta D^{(n)} = 0$ and $\delta$ is linear. Thus, solving Eq. (2.6) is equivalent to solving $\delta\Omega_i^{(n+1)} = -D_i^{(n)}$ for all $i$, as

$$\delta\Omega_i^{(n+1)} = -D_i^{(n)} \Rightarrow \delta\Omega^{(n+1)} = \delta\sum_i \Omega_i^{(n+1)} = \sum_i \delta\Omega_i^{(n+1)} = -\sum_i D_i^{(n)} = -D^{(n)}. \tag{3.5}$$

Now, since

$$\delta\sigma D_i^{(n)} = (\delta\sigma + \sigma\delta)D_i^{(n)} = N_i D_i^{(n)}, \tag{3.6}$$

as $\delta D_i^{(n)} = 0$, one sees that

$$\Omega_i^{(n+1)} = \frac{1}{N_i}\sigma D_i^{(n)} \tag{3.7}$$

is effectively a solution at degree $i$.

One can then state that for a contracting homotopy $\sigma$ for $N$,

$$\Omega^{(n+1)} = \sum_i \frac{1}{N_i}\sigma D_i^{(n)} \tag{3.8}$$

is a solution of the equation (2.6).

A particular contracting operator $\sigma$ is given by

$$\sigma = \frac{\overleftarrow{\partial}}{\partial C_a}\mathcal{P}_a \tag{3.9}$$

where the derivative with respect to the constraints is defined since the expressions considered depend only on the phase space variables through the constraints.

This definition of the contracting operator corresponds to a particular choice of the values of the parameters $\alpha_i$. Other choices would result in other partitions of the set of expressions considered, and other contracting operators.

The BRST charge resulting from the current choice of contracting homotopy operator is invariant under linear redefinitions of the constraints:

$$C_a \to C'_a = A_a{}^b C_b \tag{3.10}$$

with the appropriate redefinitions of the ghosts and ghost momenta:

$$\begin{aligned} \mathcal{P}_a &\to \mathcal{P}'_a = A_a{}^b \mathcal{P}_b \\ \eta^a &\to \eta'^a = (A^{-1})_b{}^a \eta^b. \end{aligned} \tag{3.11}$$

One is ready now to describe the two programs computing the BRST charge. They are based on the two choices of constants $\alpha_i$ 's described above. On the other hand, one is written in the algebraic mode of `REDUCE`, the other is written in its symbolic mode. A subsidiary usefulness of writing two programs is to control the validity of the calculations. Apart from the algorithms, their originality does not come from the handling of anticommuting variables (the ghost and ghost momenta) but from the extended use of dummy indices to represent the various expressions. Thanks to that, one is able to apply them without reference to the number of constraints and, of course, the complexity of the computations is also independent on this number. Expressing the brackets $\{\eta^a C_a, X\}$ for example involves as many calculations as there are constraints if the summation is explicit. On the other hand, non explicit summations on dummy variables allow the treatment of generic cases. However, the problem is now to achieve full simplification of polynomials. Outside the context of tensor algebra, there exist no package which can do that except the package `DUMMY`[5] recently created by one of the authors. Both programs also use the package `ASSIST`[6] but they use different functionalities included in it.

### 3.1. Program I

**Input**:

- Algebra of constraints Eq. (2.2)
- $\Omega^{(0)}$ Eq. (2.5)

**Output**: $\Omega^{(n)}$

The process of computations is:

$k = 0$
LOOP:
compute $D^{(k)}$ from Eq. (2.8)
extract the corresponding $Z$ from Eq. (2.9)
construct $\Omega^{(k+1)}$ from Eqs. (2.12) and (2.10)
arbitrary coefficients left in solution optionally fixed as explained below
if $k = n - 1$ then return $\Omega^{(n)}$ else k:=k+1 go to LOOP

In the program one has to handle polynomials of the type

$$\sum a^{a_1 \dots a_l c_1 \dots c_n}_{b_1 \dots b_m} C_{a_1} \dots C_{a_l} \eta^{b_1} \dots \eta^{b_m} \mathcal{P}_{c_1} \dots \mathcal{P}_{c_n} \tag{3.12}$$

where $\eta$ and $\mathcal{P}$ are odd Grassmann variables and the $a$'s stand for $Z$ or $K$. It is useful to consider the ghost and ghost momenta products as antisymmetric operators :

$$\eta^{b_1} \dots \eta^{b_m} = \eta^{b_1 \dots b_m}, \quad \mathcal{P}_{c_1} \dots \mathcal{P}_{c_n} = \mathcal{P}_{c_1 \dots c_n}. \tag{3.13}$$

#### a. calculation of $D$

Brackets of expressions like (3.12) can be decomposed into two parts, the constraint brackets

$$\sum \sum a^{a_1 \dots a_l c_1 \dots c_n}_{b_1 \dots b_m} b^{a'_1 \dots a'_{l'} c'_1 \dots c'_{n'}}_{b'_1 \dots b'_{m'}} \{C_{a_1} \dots C_{a_l}, C_{a'_1} \dots C'_{a_{l'}}\}_C$$

$$\eta^{b_1\ldots b_m b'_1\ldots b'_{m'}}\mathcal{P}_{c_1\ldots c_n c'_1\ldots c'_{n'}}(-1)^{nm'} \tag{3.14}$$

and the ghost brackets

$$\sum\sum a^{a_1\ldots a_l c_1\ldots c_n}_{b_1\ldots b_m} b^{a'_1\ldots a'_{l'} c'_1\ldots c'_{n'}}_{b'_1\ldots b'_{m'}} C_{a_1}\ldots C_{a_l}C_{a'_1}\ldots C_{a_{l'}}$$

$$\{\eta^{b_1\ldots b_m}\mathcal{P}_{c_1\ldots c_n},\eta^{b'_1\ldots b'_{m'}}\mathcal{P}_{c'_1\ldots c'_{n'}}\}. \tag{3.15}$$

In both cases, the time of calculation is considerably shortened if either the $C$'s or the ghosts or both factorize. In practice, a factorization of the ghosts is more frequent than a factorization of the constraints.

Using the Leibniz rule, the constraint brackets are easily computed from the input.

The ghost brackets implement the formula

$$\left\{\eta^{a_1\ldots a_{i_1}}\mathcal{P}_{b_1\ldots a_{i_2}},\eta^{a'_1\ldots a'_{j_1}}\mathcal{P}_{b'_1\ldots b'_{j_2}}\right\} =$$

$$\left[\sum_{k=1}^{i_1}\sum_{l=1}^{j_2}\eta^{a_1\ldots a_{k-1}a_{k+1}\ldots a_{i_1}a'_1\ldots a'_{j_1}}\mathcal{P}_{b_1\ldots b_{i_2}b'_1\ldots b'_{l-1}b'_{l+1}\ldots b'_{j_2}}(-1)^{k+l+1+i_2j_1}\delta^{a_k}_{b'_l}+\right.$$

$$\left.\sum_{k=1}^{j_1}\sum_{l=1}^{i_2}\eta^{a_1\ldots a_{i_1}a'_1\ldots a'_{k-1}a'_{k+1}\ldots a'_{j_1}}\mathcal{P}_{b_1\ldots b_{l-1}b_{l+1}\ldots b_{i_2}b'_1\ldots b'_{j_2}}\delta^{a'_k}_{b_l}(-1)^{k+l+j_1j_2+i_1i_2+i_2j_1}\right]. \tag{3.16}$$

This formula can be checked by any program dealing with anticommutating variables. The $D$ function is obtained from the bracket calculation. Because it involves many terms with the same structure, the use of the function `CANONICAL` from the package `DUMMY` is essential to simplify it. Because $\delta D^{(n)}=0$, one checks this vanishing for each value of $n$. It is interesting to note that, in all the considered examples, $\delta D^{(1)}=0$ gives all the Jacobi identities for the structure constants of the algebra.

**b. determination of the $Z$ function**

Because the result for D is obtained from `CANONICAL`, an antisymmetrization over the ghost indices is necessary to extract $Z$. After this antisymmetrization, each monomoial is divided by the constraints and the ghost momentum operator. This gives $Z$.

**c. construction of the BRST operator**

It is constructed from Eq. (2.12) where the arbitrary $G$'s are built from $Z$ by a further antisymmetrization and multiplication by arbitrary coefficients. These appear in the output but can also be fixed if one requires the number of terms appearing in the output to be minimal.

### 3.2. Program II

The input and output for the second program are identical to that of the first program: **Input**:

- Algebra of constraints Eq. (2.2)
- $\Omega^{(0)}$ Eq. (2.5)

**Output**: $\Omega^{(n)}$

The process of computation is also similar, as it follows the standard steps described earlier.

The BRST charge computed here differs from that returned by the first program in the choice of values for the coefficients $\alpha$. We have indeed chosen to adopt here the contracting operator (3.9) which yields a BRST charge invariant under linear transformations of the constraints.

This choice induces a larger number of terms in the expression of $\Omega$, and thus requires more computer resources. It was therefore necessary to work in the symbolic mode of REDUCE.

A further particularity of this program compared to the previous one lies in its handling of the Jacobi identity. Indeed, these were useful in reducing as much as possible the number of terms in the expressions without changing the contracting homotopy operator. We have built a procedure returning a normal form of polynomials with respect to the Jacobi identity for the particular algebras studied. This enabled us not only to reduce the size of the expressions, but also to check the validity of the results returned.

It should be noted however that this handling of side relations is very time consuming, and requires *ad hoc* tweaking for each new algebra.

Finally, this program has been used to study a partial classification of polynomial Poisson structures[7].

## 4. Results

Various algebras have been considered

#### Usual linear Lie algebras

The constraint algebra is given by

$$\{C_a, C_b\} = f_{ab}{}^c C_c. \tag{4.1}$$

Use of both programs gives

$$\Omega^{(1)} = \frac{1}{2} f_{ab}{}^c \eta^{ab} \mathcal{P}_c. \tag{4.2}$$

Computing $D^{(1)}$ and checking the $\delta D^{(1)} = 0$ lead to the Jacobi identity

$$f^c_{[ab}\, f^e_{d]c} = 0. \tag{4.3}$$

This leads to $D^{(1)} = 0$ and stops the construction because all the added contributions vanish.

**Self-reproducing algebras**

Let us consider the algebra

$$\{C_a, C_b\} = T_{ab} C_a C_b \tag{4.4}$$

where no summation over a,b is involved. In this section there is no summation over repeated indices. Such an algebra is characterized by the fact that Jacobi identities are trivial. At each order from the second one, an arbitrary coefficient is generated. The number of handled terms increase quickly. Fortunately, a particular choice of the arbitrary coefficient $\alpha_n$ occuring at the order $n$ by

$$\alpha_n = -\frac{n+1}{n} \tag{4.5}$$

considerably simplifies the result and the BRST operator can be obtained with program I in a closed form.

$$\Omega = \sum_{n=0}^{N-1} \frac{(-1)^{\frac{n(n-1)}{2}}}{2^n n!} \sum_b (K_b)^n C_b \eta^b \tag{4.6}$$

where

$$K_b = \sum_a \eta^a \mathcal{P}_a T_{ab} \tag{4.7}$$

The program is also able to compute $\Omega$ for arbitrary coefficients. An example of output is given in Appendix A. The expressions are much more complicated but they allow to check the compatibility of the calculations made by both programs.

**Pure quadratic algebras**

The BRST operator has been computed for the constraint algebra

$$\{C_a, C_b\} = d_{ab}^{cd} C_c C_d \tag{4.8}$$

along the same lines as above. One gets

$$\Omega^{(1)} = \frac{1}{2} d_{ab}^{cd} \eta^{ab} \mathcal{P}_c C_d. \tag{4.9}$$

The vanishing of $\delta D^{(1)}$ gives the Jacobi identity

$$d_{[ab}^{c\{g} \, d_{d]c}^{ef\}} = 0. \tag{4.10}$$

The calculation of $\Omega^{(2)}$ gives

$$\Omega^{(2)} = \frac{1}{6} d_{ab}^{cd} d_{ec}^{fg} \eta^{abc} \mathcal{P}_{df} C_g. \tag{4.11}$$

Program II allows to easily compute $\Omega$ up to order six.

**Mixed linear and quadratic algebras**

The following algebra

$$\{C_a, C_b\} = f_{ab}{}^c C_c + d_{ab}^{cd} C_c C_d \tag{4.12}$$

with

$$d_{ab}^{cd} d_{ce}^{fg} = 0 \tag{4.13}$$

is an extension of an algebra studied by Schoutens, Sevrin and Van Niewenhuizen[4]. Applying the programs, one gets

$$\Omega^{(1)} = \frac{1}{2}\left(f_{ab}{}^c \eta^{ab} \mathcal{P}_c + d_{ab}^{cd} \eta^{ab} \mathcal{P}_c C_d\right). \tag{4.14}$$

The vanishing of $\delta D^{(1)}$ again leads to the Jacobi identities

$$f_{[ab}{}^c f_{d]c}^e = 0, \tag{4.15}$$

$$f_{[ab}{}^c d_{d]c}^{ef} + d_{[ab}^{ce} f_{d]c}{}^f + d_{[ab}^{cf} f_{d]c}{}^e = 0. \tag{4.16}$$

Together with (4.13), they imply $D^{(1)} = 0$ and $\Omega^{(2)} = 0$.

$\Omega^{(3)}$ is given by

$$\Omega^{(3)} = \frac{1}{24} d_{ab}^{pe} d_{cd}^{qf} f_{pq}{}^g \eta^{abcd} \mathcal{P}_{efg}. \tag{4.17}$$

One can note that here and in the previous example, no arbirary coefficient is involved. The condition (4.13) implies the vanishing of higher orders. If this condition is released one can again compute easily $\Omega$ up to order six with program II.

**An example of cubic algebra**

The cubic algebra generated by the $C$'s

$$\{C_{d_1}, C_{d_2}\} = f_{d_1 d_2}^{d_3} C_{d_3} + D_{d_1 d_2}^{d_3 d_4} C_{d_3} C_{d_4} + E_{d_1 d_2}^{d_3 d_4 d_5} C_{d_3} C_{d_4} C_{d_5} \tag{4.18}$$

is an extension of the wellknown spin 4 algebra. The various structure constants $f, D, E$ are antisymmetric over their lower indices and symmetric over their upper indices. They satisfy the Jacobi identities :

$$f_{[ab}{}^c f_{d]c}^e = 0, \tag{4.19}$$

$$f_{[ab}{}^c d_{d]c}^{ef} + D_{[ab}^{ce} f_{d]c}{}^f + D_{[ab}^{cf} f_{d]c}{}^e = 0, \tag{4.20}$$

$$2D_{d_7[d_4}^{\{d_1 d_2} D_{d_5 d_6]}^{d_3\}d_7} + E_{d_7[d_4}^{\{d_1 d_2 d_3\}} f_{d_5 d_6]}^{d_7} + 3 f_{d_7[d_4}^{\{d_1} E_{d_5 d_6]}^{d_2 d_3\}d_7} = 0, \tag{4.21}$$

$$3D_{d_1[d_2}^{\{d_3 d_4} E_{d_7 d_8]}^{d_5 d_6\}d_1} + 2E_{d_1[d_2}^{\{d_3 d_4 d_5} D_{d_7 d_8]}^{d_6\}d_1} = 0, \tag{4.22}$$

$$E_{[d_6 d_7}^{d_8\{d_1 d_2} E_{d_9]d_8}^{d_3 d_4 d_5\}} = 0. \tag{4.23}$$

The BRST charge for this algebra is given to order six in Appendix B.

## 5. Conclusions.

As shown by the results in the previous sections, the new programs allow the computation of the BRST operator when the algebras of constraints are more complicated than the usual linear algebras. We recall that quadratic algebras have been considered recently in the framework of the study of superconformal field theories[4] Program I is written in the algebraic mode of `REDUCE` . It is, of course, less efficient than the program written in symbolic mode but is still quite able to make the most relevant computations in a reasonable time. Its main limitation is the fact that it does not take properly into account the Jacobi identities. For instance, the calculation of the first nontrivial term after $\Omega^{(1)}$ is incorrect by a numerical factor $\frac{1}{3}$. The necessity to take into account the Jacobi identities can be implemented in program I but program II fulfills that job and, so it does not look to be worth the task. As far as this aspect is concerned, it should be stressed that it is because of the high level of symmetry of Poisson algebra structures that this was possible and it is not claimed that the algorithm is efficient. Finally, the specificity of program I with respect to program II lies in the calculation of the BRST operator in self-reproducing algebras. The result can indeed be written in a more compact form than the choice of the contracting homotopy made in program II allows to. Programs are available, on request, by electronic mail at the following address caprasse@vm1.ulg.ac.be.

## References


1. M. Henneaux and C. Teitelboim, *Quantization of Gauge Systems*, Princeton University Press, New Jersey (1993).
2. A.C. Hearn, *"User's Manual", Version 3.5* RAND, Santa Monica, CA 90407-2138J. (1993).

3 K. Schoutens, A. Sevrin, P. van Nieuwenhuizen, *Commun. Math. Phys.* **124***, 87 (1989)*

4 Z. Khviengia and E. Sezgin, *Phys. Lett.* **B 326***, 243 (1994)*

5 A. Dresse *"DUMMY.RED"* `REDUCE` library (1994).

6 H. Caprasse *"ASSIST.RED"* `REDUCE` library (1993).

7 A. Dresse and M. Henneaux *J. Math. Phys.* **35 vol. 3***, 1334 (1994)*

8 W. Antweiller *"TRI.RED"* `REDUCE` library (1993).

**Appendix A**

$omega(3)$ computed from program I for arbitrary parameters $alpha2, alpha3$ :

$omega(3) := brstconstr(3, any);$

$$\begin{aligned} omega(3) := (&alop(s1,s2) * alop(s1,s3) * eta(s1,s2,s3,s4) * \\ &(-3 * alop(s1,s4) * contr(s1) * prond(s2,s3,s4) * alpha2 * alpha3 \\ &- 3 * alop(s1,s4) * contr(s1) * prond(s2,s3,s4) * alpha3 \\ &+ 9 * alop(s1,s4) * contr(s4) * prond(s1,s2,s3) * alpha2 * alpha3 \\ &+ 12 * alop(s1,s4) * contr(s4) * prond(s1,s2,s3) * alpha2 \\ &+ 9 * alop(s1,s4) * contr(s4) * prond(s1,s2,s3) * alpha3 \\ &+ 12 * alop(s1,s4) * contr(s4) * prond(s1,s2,s3) \\ &+ 56 * alop(s3,s4) * contr(s3) * prond(s1,s2,s4) * alpha2 \\ &+ 84 * alop(s3,s4) * contr(s3) * prond(s1,s2,s4) \\ &+ 8 * alop(s3,s4) * contr(s4) * prond(s1,s2,s3) * alpha2 \\ &+ 12 * alop(s3,s4) * contr(s4) * prond(s1,s2,s3)))/96\$ \end{aligned}$$

$omega(3)$ computed from program I with $alpha2 = -3/2,\ alpha3 = -4/3$ :

$omega(3) := brstconstr(3, simplify);$

$$\begin{aligned} omega(3) := (&alop(s1,s2) * alop(s1,s3) * alop(s1,s4) * alop(s1,s5) * alop(s1,s6) * \\ &contr(s1) * eta(s1,s2,s3,s4,s5,s6) * prond(s2,s3,s4,s5,s6))/3840 \end{aligned}$$

## Appendix B

An example of the output from Program II for the cubic algebra follows. The formula below is a direct TEX output of package `TRI`[8] of `REDUCE`. It should be clear that we can compute $\Omega$ to higher orders.

$$
\begin{aligned}
\Omega(6) = \big(&\eta^{d_1}\,\eta^{d_2}\,\eta^{d_3}\,\eta^{d_4}\,\eta^{d_5}\,\eta^{d_6}\,\eta^{d_7}\,\mathcal{P}_{d_8}\,\mathcal{P}_{d_9}\,\mathcal{P}_{d_{10}}\,\mathcal{P}_{d_{11}}\,\mathcal{P}_{d_{12}}\,\mathcal{P}_{d_{13}}\\
&\Big(864\,f^{d_9}_{d_{14}d_2}\,f^{d_{10}}_{d_{15}d_{16}}\,f^{d_{11}}_{d_{17}d_{18}}\,C_{d_{19}}\,E^{d_{15}d_{17}d_{12}}_{d_3d_4}\,E^{d_{18}d_{19}d_{13}}_{d_5d_6}\\
&E^{d_{14}d_{16}d_8}_{d_7d_1} + 864\,f^{d_9}_{d_{14}d_{15}}\,f^{d_{10}}_{d_{16}d_2}\,f^{d_{11}}_{d_{17}d_{18}}\,C_{d_{19}}\,E^{d_{17}d_{19}d_{12}}_{d_3d_4}\\
&E^{d_{14}d_{16}d_{13}}_{d_5d_6}\,E^{d_{15}d_{18}d_8}_{d_7d_1} - 126\,f^{d_9}_{d_{14}d_{15}}\,f^{d_{10}}_{d_{16}d_{17}}\,D^{d_{18}d_{12}}_{d_3d_4}\\
&D^{d_{16}d_{11}}_{d_{18}d_2}\,C_{d_{19}}\,E^{d_{14}d_{17}d_{13}}_{d_5d_6}\,E^{d_{15}d_{19}d_8}_{d_7d_1} - 63\,f^{d_9}_{d_{14}d_{15}}\,D^{d_{16}d_{10}}_{d_2d_3}\\
&D^{d_{17}d_{11}}_{d_4d_5}\,D^{d_{18}d_{12}}_{d_{16}d_{17}}\,D^{d_{14}d_{13}}_{d_{18}d_6}\,C_{d_{19}}\,E^{d_{15}d_{19}d_8}_{d_7d_1} - 42\,f^{d_9}_{d_{14}d_{15}}\\
&D^{d_{16}d_{10}}_{d_2d_3}\,D^{d_{18}d_{11}}_{d_4d_5}\,D^{d_{17}d_{12}}_{d_{16}d_6}\,D^{d_{14}d_{13}}_{d_{18}d_{17}}\,C_{d_{19}}\,E^{d_{15}d_{19}d_8}_{d_7d_1} - 168\\
&f^{d_9}_{d_{14}d_{15}}\,D^{d_{16}d_{10}}_{d_2d_3}\,D^{d_{17}d_{11}}_{d_{16}d_4}\,D^{d_{18}d_{12}}_{d_{17}d_5}\,D^{d_{14}d_{13}}_{d_{18}d_6}\,C_{d_{19}}\,E^{d_{15}d_{19}d_8}_{d_7d_1}\\
&+ 84\,D^{d_{14}d_{11}}_{d_1d_2}\,D^{d_{17}d_{12}}_{d_3d_4}\,D^{d_{18}d_{13}}_{d_5d_6}\,D^{d_{16}d_{10}}_{d_{14}d_{15}}\,D^{d_{19}d_9}_{d_{17}d_{18}}\,D^{d_{15}d_8}_{d_{19}d_7}\,C_{d_{16}}\\
&+ 56\,D^{d_{14}d_{11}}_{d_1d_2}\,D^{d_{17}d_{12}}_{d_3d_4}\,D^{d_{19}d_{13}}_{d_5d_6}\,D^{d_{16}d_{10}}_{d_{14}d_{15}}\,D^{d_{15}d_9}_{d_{17}d_{18}}\,D^{d_{18}d_8}_{d_{19}d_7}\,C_{d_{16}}\\
&+ 168\,D^{d_{17}d_{11}}_{d_1d_2}\,D^{d_{18}d_{12}}_{d_3d_4}\,D^{d_{19}d_{13}}_{d_5d_6}\,D^{d_{16}d_{10}}_{d_{14}d_{15}}\,D^{d_{15}d_9}_{d_{17}d_{18}}\,D^{d_{14}d_8}_{d_{19}d_7}\,C_{d_{16}}\\
&+ 224\,D^{d_{14}d_{10}}_{d_2d_3}\,D^{d_{17}d_{11}}_{d_4d_5}\,D^{d_{16}d_9}_{d_{14}d_{15}}\,D^{d_{18}d_8}_{d_{17}d_1}\,D^{d_{19}d_{13}}_{d_{18}d_7}\,D^{d_{15}d_{12}}_{d_{19}d_6}\,C_{d_{16}}\\
&+ 84\,D^{d_{16}d_{11}}_{d_2d_3}\,D^{d_{17}d_{12}}_{d_4d_5}\,D^{d_{15}d_{10}}_{d_{14}d_1}\,D^{d_{18}d_9}_{d_{16}d_{17}}\,D^{d_{19}d_8}_{d_{18}d_7}\,D^{d_{14}d_{13}}_{d_{19}d_6}\\
&C_{d_{15}} - 126\,D^{d_{16}d_{11}}_{d_2d_3}\,D^{d_{18}d_{12}}_{d_4d_5}\,D^{d_{19}d_{13}}_{d_6d_7}\,D^{d_{15}d_{10}}_{d_{14}d_1}\,D^{d_{14}d_8}_{d_{16}d_{17}}\\
&D^{d_{17}d_9}_{d_{18}d_{19}}\,C_{d_{15}} - 140\,D^{d_{16}d_{11}}_{d_2d_3}\,D^{d_{18}d_{12}}_{d_4d_5}\,D^{d_{15}d_{10}}_{d_{14}d_1}\,D^{d_{17}d_{13}}_{d_{16}d_6}\\
&D^{d_{14}d_9}_{d_{17}d_{19}}\,D^{d_{19}d_8}_{d_{18}d_7}\,C_{d_{15}} + 336\,D^{d_{16}d_{11}}_{d_2d_3}\,D^{d_{18}d_{12}}_{d_4d_5}\,D^{d_{15}d_{10}}_{d_{14}d_1}\\
&D^{d_{14}d_9}_{d_{16}d_{17}}\,D^{d_{19}d_8}_{d_{18}d_7}\,D^{d_{17}d_{13}}_{d_{19}d_6}\,C_{d_{15}} + 56\,D^{d_{16}d_{11}}_{d_2d_3}\,D^{d_{19}d_{12}}_{d_4d_5}\\
&D^{d_{15}d_{10}}_{d_{14}d_1}\,D^{d_{18}d_9}_{d_{16}d_{17}}\,D^{d_{14}d_{13}}_{d_{18}d_6}\,D^{d_{17}d_8}_{d_{19}d_7}\,C_{d_{15}} - 56\,D^{d_{17}d_{10}}_{d_2d_3}\,D^{d_{18}d_{11}}_{d_4d_5}\\
&D^{d_{16}d_9}_{d_{14}d_{15}}\,D^{d_{15}d_{13}}_{d_{17}d_7}\,D^{d_{19}d_8}_{d_{18}d_1}\,D^{d_{14}d_{12}}_{d_{19}d_6}\,C_{d_{16}} - 224\,D^{d_{16}d_{10}}_{d_3d_4}\\
&D^{d_{15}d_9}_{d_{14}d_2}\,D^{d_{17}d_8}_{d_{16}d_1}\,D^{d_{19}d_{13}}_{d_{17}d_7}\,D^{d_{14}d_{11}}_{d_{18}d_5}\,D^{d_{18}d_{12}}_{d_{19}d_6}\,C_{d_{15}}\Big)\Big)\,/211680
\end{aligned}
$$